

Sub
Pa

program; and

performing an exception handling for writing an address which is obtained based on said label onto said first program when the judging determines that the address is not obtained from the label.

16. (AS NEW) A computer program for making a computer function as a program translating device when the computer executes the program, said program comprising:

encoding a source code of a target program of one of an interpreter and a compiler, said target program being described by using a label; and

performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address.--

REMARKS

In the Office Action mailed November 20, 2002, the drawings were objected to under 37 C.F.R. 1.84(p)(4) and 1.83(a), the specification was objected to for informalities, and claims 1-7 were rejected under 35 U.S.C. 102(b) as being anticipated by Nakamura (Japanese Patent No. JP405100864A). The foregoing objections and rejections are respectfully traversed.

In accordance with the foregoing, new claims 8-16 are added, and the specification and claim 1, 4, 5, 6, and 7 have been amended. Claims 1-16 are pending and under consideration.

Care has been exercised to avoid the introduction of new matter. A Version with Markings to Show Changes Made to the specification and the claims is included herewith.

Objections to the Drawings/Objections to the Specification

The specification is amended to overcome the objections to the drawings and to the specification. With respect to "PD" disclosed on page 6 at line 4 of the subject specification, the attorney for the applicants is advised that "PD" was known at the time of the present invention in Japan, and means "Phase change rewritable Disk".

Withdrawal of the objections to the drawings and to the specification is respectfully requested.

Rejections of Claims 1-7

Nakamura discusses that "A priority decision part 9 converting the content of a 'label part' in an address which is to be executed next in a format on a label reference table into the content of the 'label part' of a micro instruction whose priority is the highest by viewing the label reference table of the format translated into a machine word is provided for the micro program assembler" (refer to the ABSTRACT of Nakamura). Thus, Nakamura converts a content of a label reference table.

Nakamura's function of converting the content of the label reference table does not correspond to, or suggest, exception handling as in the present invention. Also in contrast to the present invention, Nakamura does not discuss or suggest that exception handling changes a flow of a program.

In contrast to Nakamura, one feature of the present invention not discussed or suggested by Nakamura is "performing an exception handling for writing an address which is obtained based on said label onto said program", as recited in each of claims 1, 4, 7, 8, 9, 10, 14, and 15 of the present application (using the recitation of claim 1 as an example).

In addition, as recited in each of claims 11, 12, 13, and 16 of the present application, another feature of the present invention not discussed or suggested by Nakamura is "performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address" (using the recitation of claim 11 as an example).

Dependent claims 2, 3, 5, and 6 recite patentably distinguishing features of their own. For example, claim 2/1 recites "said label address translating unit uses a table showing a relationship between the label and the address".

Withdrawal of the rejections of claims 1-7 and allowance of new claims 8-16 is respectfully requested.

There being no further outstanding objections or rejections, it is submitted that the

application is in condition for allowance. An early action to that effect is courteously solicited.


Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: April 21, 2003

By: 
Gene M. Garner II
Registration No. 34,172

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

Please AMEND the paragraph beginning at page 6, line 2, as follows:

The readably-by-computer medium may embrace, e.g., a CD-ROM, a magnetic disk such as a floppy disk, a magneto-optic disk such as an MO (magneto-optic), and an optical disk such as a PD (Phase change rewritable Disk).

Please AMEND the paragraph beginning at page 7, line 1, as follows:

To start with, an architecture of a label address translating device in the embodiment of the present invention will be explained referring to FIGS. 2 through 5. FIG. 1 is a diagram showing a construction of a label address translating device 10. FIG. 2 is a diagram showing one example of a source code of a program 20 shown in FIG. 1. FIG. 3 is a diagram showing one example of a code obtained by compilation of the program. FIG. 4 is a diagram one example of a label table [24] 32 illustrated in FIG. 1. FIG. 5 is a diagram showing one example of a code in which an address is rewritten by a label address translating unit 18 shown in FIG. 1.

Please AMEND the paragraph beginning at page 8, line 4, as follows:

A source code 28 of the program 20 is described by use of a label name (LABEL_B) in a predetermined language, e.g., an interpreter language (see FIG. 2). Further, the source code 28 is encoded by an interpreter contained in the source code 28 [(see FIG. 3)] (see FIG. 2). The encoded program 20 is developed on the main memory 14 and executed by the CPU 12.

Please AMEND the paragraph beginning at page 10, line 15, as follows:

According to the label address translating device 10, when recognizing the invalid data during the execution of the program 20 by the program execution unit 16, the label address translating unit 18 starts up the exception handler 22. Subsequently, the exception handler 22 obtains the effective address from the label defined in the program 20 by referring to [the label

tables 24, 32,] the label table 32 and sets the thus obtained effective address in the program 20. Namely, the CPU 12, upon recognizing the invalid data, executes the exception handling and the normal processes thereafter.

Please AMEND the paragraph beginning at page 11, line 25, and continuing through page 12, line 7, as follows:

As shown in FIG. 7, the CPU 12, upon recognizing the invalid data during the execution of the program 20, starts up the exception handler 22 (S11). The exception handler 22 refers to the label [tables 24, 31] table 32, thereby obtaining the effective address, wherein the address of the command in which the invalid data is described (or the label ID of this command) serves as a key (S12). Subsequently, the CPU 12 rewrites the address described in that command (which is, e.g., an address for storing the arithmetic result) into the effective address from the invalid data (S13).

Please AMEND the paragraph beginning at page 12, line 16, as follows:

According to the label address translating device 10 in the embodiment discussed above, the label [tables 24, 32] table 32 showing the relationship between the labels and the effective addresses are referred to, whereby each of the labels described in the program 20 is rewritten into the effective address. The program 20 as a target for processing eventually falls into a state where the effective address is rewritten therein instead of the label. Hence, in the execution (compilation) of the program 20 after being rewritten, since there is no necessity for translating the label into the effective address, it is possible to omit the process of translating the label into the effective address.

IN THE CLAIMS:

Please AMEND the following claims:

1. (ONCE AMENDED) A label address translating device for obtaining an address from

a label described in a program, comprising:

a program processing unit judging whether or not the address is obtained from the label when in the processing of the program; and

a label address translating unit [obtaining,] performing an exception handling for writing an address which is obtained based on said label onto said program when said program processing unit judges that the address is not obtained from the label[, the address from the label in an exception handling, and writing the address to the program].

4. (ONCE AMENDED) A label address translating method of obtaining an address from a label described in a program, comprising:

[a first step of] processing the program;

[a second step of] judging whether or not the address is obtained from the label when the processing the program in said [first step] processing; and

[a third step of obtaining,] performing an exception handling for writing an address which is obtained based on said label onto said program when judging in said [second step] judging that the address is not obtained from the label[, the address from the label in an exception handling; and

a fourth step of writing the address obtained in said third step to the program].

5. (ONCE AMENDED) A label address translating method according to claim 4, wherein said [third step] performing involves reading the address to be obtained from a table showing a relationship between the label and the address.

6. (ONCE AMENDED) A label address translating method according to claim 4, wherein said [second step involves] judging includes judging, when the address indicated by the label is not a real address, that the address is not obtained from the label.

7. (ONCE AMENDED) A readable-by-computer medium stored with a second program for making a computer function as a label address translating device for obtaining an address from a label described in a first program, said second program comprising:

[a judging step of] judging whether or not the address is obtained from the label when processing said first program;

[an obtaining step of obtaining,] performing an exception handling for writing an address which is obtained based on said label onto said first program when judging in [said judging step] that the address is not obtained from the label[, the address from the label in an exception handling; and
 a writing step of writing the address obtained in said obtaining step to said program].

Please ADD the following claims:

--8. (NEW) A label address translating device for obtaining an address from a label described in a program of one of an interpreter and a compiler, comprising:

a program processing unit judging whether or not the address is obtained from the label when in the processing of the program; and

a label address translating unit performing an exception handling for writing an address which is obtained based on said label onto said program when said program processing unit judges that the address is not obtained from the label.

9. (NEW) A label address translating method of obtaining an address from a label described in a program of one of an interpreter and a compiler, comprising:

processing the program;

judging whether or not the address is obtained from the label when the processing the program in said processing; and

performing an exception handling for writing an address which is obtained based on said label onto said program when judging in said judging that the address is not obtained from the label.

10. (NEW) A computer readable medium storing a first program making a computer function as a label address translating device for obtaining an address from a label described in a second program of one of an interpreter and a compiler when said computer executes said first program, said first program comprising:

judging whether or not the address is obtained from the label when processing said second program; and

performing an exception handling for writing an address which is obtained based on said label onto said second program when the judging determines that the address is not obtained from the label.

11. (NEW) A program translating device comprising:

an encode unit encoding a source code of a program of one of an interpreter and a compiler, said program being described by using a label; and

an exception handling unit performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address.

12. (NEW) A program translating method comprising:

encoding a source code of a program of one of an interpreter and a compiler, said program being described by using a label; and

performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address.

13. (NEW) A computer readable medium storing a translating program for making a computer function as a program translating device when the computer executes the program, said translating program comprising:

encoding a source code of a target program of one of an interpreter and a compiler, said target program being described by using a label; and

performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address.

14. (NEW) A computer program making a computer function as a label address

translating device for obtaining an address from a label described in a target program when the computer executes the program, said program comprising:

judging whether or not the address is obtained from the label when the processing said target program; and

performing an exception handling for writing an address which is obtained based on said label onto said first program when the judging determines that the address is not obtained from the label.

15. (NEW) A computer program for making a computer function as a label address translating device for obtaining an address from a label described in a target program of one of an interpreter and a compiler when the computer executes the program, said program comprising:

judging whether or not the address is obtained from the label when processing said first program; and

performing an exception handling for writing an address which is obtained based on said label onto said first program when the judging determines that the address is not obtained from the label.

16. (NEW) A computer program for making a computer function as a program translating device when the computer executes the program, said program comprising:

encoding a source code of a target program of one of an interpreter and a compiler, said target program being described by using a label; and

performing, when said encoded source code includes an invalid address, an exception handling for replacing said invalid address with a valid address.--